

SOFTWARE MANAGEMENT AT SLS

R. Krempaská, A. Al-Adwan, A. Bertrand, M. Dach, M. Grunder, M. Heiniger, C. Higgs, S. Hunt, B. Kalantari, R. Kappeler, T. Korhonen, J. Krempaský, A. Luedeke, D. Maden, W. Portmann, D. Vermeulen, D. Zimoch

The SLS Machine and Beamlines Control Group is responsible for machine and beamlines control systems. EPICS (Experimental Physics and Industrial Control System) is used for this purpose. Hundreds of control system applications running on about 200 IOCs (Input/Output Controllers) have been developed. With so many applications, it becomes increasingly difficult to maintain or to upgrade older applications, to produce new applications and to upgrade EPICS, VxWorks and Linux operating systems with minimal disruption to machine and beamlines operations. Thus software management of control system application becomes extremely important.

INTRODUCTION

The software used at SLS consists of many components and packages, and runs on heterogeneous architectures. The main effort in our software development is focused on machine and beamlines control system applications. For this purpose the EPICS [1] toolkit is used. Basic components of EPICS are a VME processor running on VxWorks operating system and OPI (Operator Interface) software running on Linux.

To ensure the functionality, consistency and stability of the SLS machine and beamlines control systems, each SLS developer has to follow predefined rules and guidelines [2].

An important part of software development is a test plan, a collection of essential and useful information for each control system application.

SLS APPLICATION TYPES

Applications at SLS can be classified into two main types. The first type comprises low-level applications, the IOC-based VME applications running on I/O controllers and containing mainly EPICS databases. The second type comprises high-level Linux based applications, such as graphical user interfaces, scripts and programs written in C, C++, etc. Each application type has its directory structure in our repository tree.

CVS DIRECTORY STRUCTURE

Software used by the Computing and Controls Group is under constant development. In order to ensure a consistent quality standard, all the software is supervised with the code management system, CVS [3]. Developers can independently get a particular version of a code from the CVS repository, work on it, and then commit it back into the repository.

The CVS root directory contains one sub-directory structure for third party source code (*PACKAGES*), e.g. VxWorks, EPICS, CDEV. Another sub-directory structure is dedicated for code written at SLS (*PROJECTS*) and a third sub-directory structure for general-purpose utilities (*UTILITIES*). Each unit of the machine and beamline control systems has to be placed in a

PROJECTS/subsystems area with the subsystem name.

THE DIRECTORY STRUCTURE

Three directory structures, */devl*, */work* and */prod* are used. For development and initial testing, the public PSI network is used. The final control systems tests and releases into production are done on the private SLS network. */devl* is an area where initial application testing takes place on the PSI network. */work* is an area where integration testing is carried out on the SLS network. During machine maintenance, IOCs can boot from */work*. */prod* is an area where the operational code is kept for execution on the SLS network. During User Operations, all IOCs boot from */prod*. Before installation, the application has to be "checked-in" and "tagged". A tag is a CVS method which gives a label to the collection of files belonging to the application. To install code into */work* or */prod*, we use a script which extracts the "tagged" version from CVS and copies the files to the correct locations. Installation into */prod* requires a test plan [4].

SUBSYSTEM DEVELOPMENT CYCLE

The software development process consists generally of following stages: gathering and analyzing requirements, application specifications, development management, comprising design, implementation and testing and application release. Fig. 1) shows SLS machine and beamlines control systems development and release stages.

Before starting development, a test plan has to be created which becomes a part of each subsystem's documentation. Then the files to be modified are checked out from CVS, (thus preserving the directory structure), into a local working directory. In this local working directory all code modifications are done. Initial testing is encouraged from this local working directory. If this is not possible, then the code is copied into */devl* for testing.

During machine maintenance, the modifications may be tested from */devl* (PSI network) or */work* (SLS network). Before final installation into */prod*, all changes must be committed back to CVS. An installation script

will then release the application to the correct location in */prod*. Finally “live” testing is performed from */prod*. After successful testing, the test plan is updated with the test results.

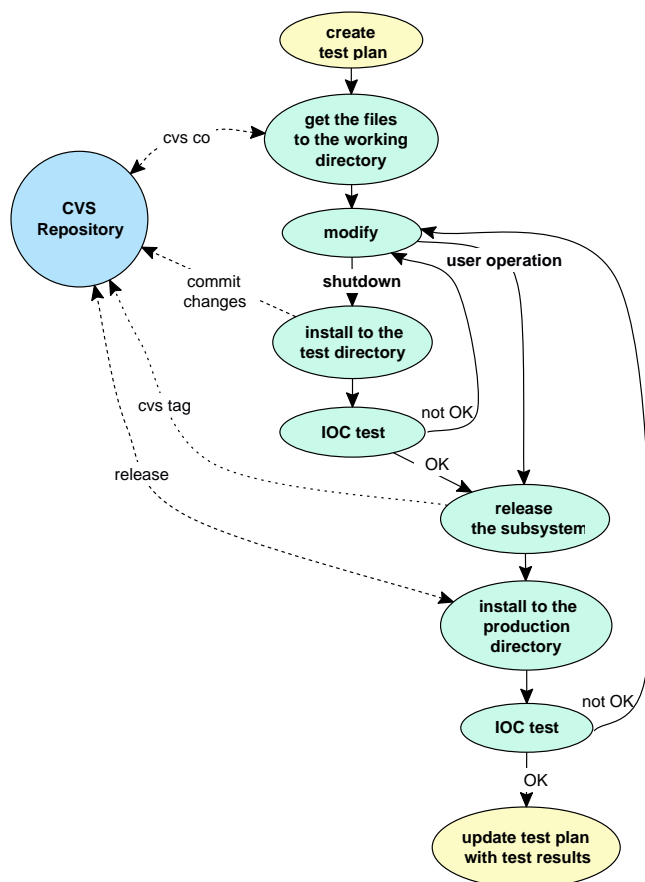


Fig. 1: Schematic diagram of subsystem development, installation and testing cycle.

TEST PLAN

A test plan [5] is a collection of useful information to keep track of control system software. Test plan documents govern the installation, testing and release of new applications. A test plan must contain the following information; subsystem name and version, time required for test with and without beam, special test requirements, reasons for change, benefits and/or risks, procedures for installation, testing and rollback and test results.

A Web based application has been developed to create, modify and view software test plans. For user interface development PHP has been used, (an HTML-embedded scripting language). Subsystem developers can create a new test plan. An author of a test plan can modify or remove it.

All the information entered via the Web is stored in an Oracle Database. This is convenient, since some information, e.g. list of IOC crates for a particular subsystem, IOC boot parameters, etc. can be automatically inserted, maintained and reported.

FUTURE IMPROVEMENTS AND GOALS

The subsystem development and installation cycle has been tested and is being used for SLS machine control system applications. Since the beamline subsystems presently use a different directory structure, work is still required to unify the mechanism.

The management of applications running on Microsoft Windows Operating Systems has to be taken into consideration. Unfortunately, in some special cases, developers have to use MS Windows. This is usually the case in developing software for CCD (Charged Coupled Device) cameras. In this case, the API (Application Programming Interface) supported by the CCD camera manufacturer runs only under Windows.

Test plan documents have been implemented for low level applications (IOCs). Test plans for high level applications, running on Linux and test plans for upgrade of EPICS, VxWorks and Linux operating systems have to be defined and implemented. Further improvements on the system which supplies information to test plan documents are planned. Using common *Makefile* templates, the developer should have the possibility to include more options in the standard release script, (e.g. compile the checked-out code from CVS, etc).

REFERENCES

- [1] S. Hunt *et al.*, *Status of the SLS Control System*, PSI Scientific Report 2001, VII.
- [2] <http://www.sls.psi.ch/controls/help/howto/sw-guidelines.html>
- [3] P. Cederquist *et al.*, *Version Management with CVS*
- [4] K. White, private communications
- [5] <http://www.sls.psi.ch/controls/help/howto/tpldevelopment.html>
- [6] S. Schaffner *et al.*, *Tools for Application Management at Jefferson Lab*, ICALEPS 1999, Trieste
- [7] http://www.sls.psi.ch/controls/infrastructure/epicsdb_iocboot.html
- [8] http://www.sls.psi.ch/controls/intranet/software/sls_software_uguide_2.html